

**Software Development for
INSAT-2E
Attitude & Orbit Control System**

*S. Udupa, Fitzgerald J. Archibald, M.M. Nayak, S. Lakshmi,
S. Balaji, V.K. Agrawal, N.K. Malik*

*Control Systems Group
ISRO Satellite Centre
Bangalore – 560 017*

Abstract

INSAT-2E is a multipurpose Geostationary Satellite. Attitude and Orbit Control System (AOCS) is a mission critical system of the satellite. This paper discusses the software development for the on-board computer of a such a mission critical system. Efforts are made to adopt standard software engineering practices within the limitation of the application environment. The performance of the software with pre and post launch of the satellite has been very much satisfactory.

1. Introduction:

INSAT-2E is a Geostationary Satellite launched on 3rd April 1999. This is the fifth satellite in the INSAT – 2 series. This carries communication payload as well as meteorological payloads. Eversince the launch of first satellite on October 1954, the progress in the area of satellite technology has been very significant. Today's satellite demands higher pointing and stability requirements. This is also true in case of INSAT-2E. The demand of high pointing and stability requirements result in complex AOCS in addition to very high reliability for long duration (12-15 years) uninterrupted mission life.

The AOCS is an integral part of the satellite. It does the function of controlling attitude errors and stability despite of several disturbances acting on the satellite dynamics. Block diagram of a typical AOCS is given in fig. 1. The AOCS consists of attitude and orbit control computer (AOCC), sensors and actuators. The AOCC receives the error and error rate signals from various sensors such as earth sensors, sun sensors and rate gyros. The received signals from the sensors are processed by the AOCC in accordance with predefined control laws using controller. The output of this controller is fed to actuators such as momentum/reaction wheels, thrusters and magnetic torquers. These actuators mounted on satellite provide required correcting torques. Thus, satellite is stabilized despite disturbance torque acting on it.

In addition to sensors and actuators, AOCC also interfaces with various other systems of the satellite. Block schematic of AOCC along with its interfaces is shown in fig. 2. The interface with telecommand gives the capability of commanding the AOCC to required mode of operation. The telemetry interface provides the required health monitoring of the AOCC. Power system provides the power to the AOCC whereas solar array drive enables AOCC to control the solar array position. Meteorological payload interfaces are required to synchronize AOCC with the payload operation for accurate imaging. More details of AOCC hardware may be found in ref [2]. The AOCC is designed using MIL-STD-1750A processor and the software is developed in Ada language for better software reliability. In this paper, we present the software design of INSAT-2E AOCC. In the following section we present the brief summary of functional requirements of the INSAT-2E AOCS operation. Then we give the design constraints, design methodology and development environment in section 3. This section is followed by the software design description in details. Here, the various components of software design are discussed. Finally, the ground as well as onboard performances of AOCC software are highlighted.

2. Functional Description:

Starting from launch pad, the satellite undergoes various phases of the mission. INSAT-2E mission profile with reference to the AOCS operation is shown in fig 3.

Initially, the satellite is kept in launch phase configuration where AOCS, only performs housekeeping monitoring functions and commanding operation. The controller operations are kept in non-active mode called suspended mode. During the launch vehicle separation phase, the AOCS automatically orients the spacecraft in sun acquisition mode where the folded solar panel mounted on spacecraft south side are pointed towards sun, called south sun acquisition mode. It is then followed by earth acquisition, to point towards earth for gyro calibration. The spacecraft orbit is then changed from transfer orbit (200 km x 36,000 km) to about 36000-km circular orbit by firing the Liquid Apogee Motor (LAM) in LAM mode. Thereafter, the on-orbit sun acquisition mode called east/west sun acquisition, earth acquisition, appendages

deployment and final on-orbit mode are followed before the spacecraft is made operational for payload operation. The spacecraft orbit keeps on drifting due to various disturbances. To correct this drift a special mode called station keeping is provided. This station-keeping mode keeps the spacecraft in proper orbit. Further details of the spacecraft operations are beyond the scope of this paper and are found in ref [8].

As mentioned above, the spacecraft goes through several modes of its operation. The AOCC keeps on acquiring sensor data at regular interval. All the telemetry and commanding operations are executed at regular interval. List of important software functions are given in Table 1 along with the periodicity, option available, and execution control option, etc.

Before going into the details of the software design aspects, we first give briefly the software application environment in which the software is to be developed and the software development methodology.

3. Software Application Environment:

The design of software is largely influenced by its application environment. For a spacecraft application the software is constrained by several factors. A suitable development methodology is needed to develop such software for such a high reliable application. Development environment also plays an important role in the software development. In this section, the design constraints, design methodology and development environment are briefly described.

3.1 Design Constraints:

To appreciate the software development methodology, it is important to know the constraints under which the software is to be developed. Some of the constraints in the development of our AOCC software are given below:

Computational Speed:

Onboard applications in spacecraft need rad-hard and highly reliable processors. Very few processors are available in this category and are not comparable to the latest available commercial processors in terms of architecture and speed.

Physical Limitation:

Weight, volume and power are always limited for a spacecraft system. A good software methodology helps in developing software with less memory usage and less computation time and also it provides with clear and simple software.

Development Support:

A good development tool is essential for developing good software. Presently we are using the in-house development tool and the tool supports the software development throughout its life cycle.

Requirements and Changes:

The methodology should be capable of tolerating changes at any stage of development. Many a times' design has to start with partial requirements. This is due to the fact that system design remains tentative till spacecraft design is completed and it will be late to start, if one has to wait till the design of spacecraft is completed. Therefore, the software structure should be flexible to accommodate changes, even at later stage till complete requirements are available.

Reliability:

The space system's software should be highly reliable as the spacecraft is irrepairable in space and it is required to function continuously for 10 to 15 years.

Susceptibility to Single Event Upsets:

Single event upset is a common phenomena in an onboard system due to cosmic radiations. The software should be designed to tolerate such upsets without any adverse impact.

3.2 Design Methodology

Fig. 4 shows the AOCC software development methodology. The integrated design approach of development of AOCC software consists of the following steps to be performed in sequence.

- Determine hardware interfaces

- Assign processes to the end functions
- Decompose the middle part
- Determine concurrency
- Determine process interfaces
- Introduce intermediary processes
- Encapsulate tasks in packages or library units
- Translate design into suitable program design language (PDL)
- Decompose large tasks further and conduct reviews

3.3 Development Environment

The AOCC software was developed on an IBM RS6000 Unix platform. TLD system supplied the compiler, assembler for 1750A, linker, debugger and simulator. Fig. 5 shows the schematic of full software development environment. After complete software development, using in-house developed Development Tool (DT) complete software can be downloaded to actual AOCC hardware and complete integrated tests can be carried out using debugger. The in-house developed DT is having a monitor program which supports function such as start up ROM emulation, memory read/write, IO operations, register read/write, break point execution, conditional break point execution, direct execution, execution through script file, interrupt execution etc. The DT was interfaced with IBM RS-6000 through RS-232 link with baud rate of 9600. The IBM RS6000 workstation with in-house developed DT was serving as full-fledged development environment for AOCC software development. This development set-up is unique and can be used for software development of different types of spacecraft.

4. AOCC Software Development :

AOCC software is mission critical on-board software since no modification or repair is possible once the spacecraft is launched. So, a systematic software development approach is used during software development. Identifying clearly the following phases of software development follows software-engineering practices.

- Requirement Analysis
- Design

- Coding
- Testing

Following gives a brief description of activities carried out during each phases of software development specifically from the point of view that software for INSAT-2E AOCC is developed using higher level language Ada.

4.1 Requirement Analysis

The AOCC software should have a real time executive which should complete execution in 32.768 msec with at least 25% margin in execution time and this execution should take place every 32.768 msec. The AOCC software requirements are classified in to four major portions.

- Mode preprocessing
- Mode processing
- Mode Post processing
- Time Synchronization

The schematic of AOCE cycle is shown in fig. 6.

Mode preprocessing:

The AOCC software configuration is mainly identified by the mode in which the AOCE is operating. In this portion activities, which are to be carried out prior to mode processing, are done. Major functions in this portion are

- Telecommand processing
- Telemetry processing
- Data Acquisition (sensor)
- Data processing (sensor)

Since the processed data is used by mode processing routine, these functions are to be carried out prior to mode processing in the time domain.

Mode Processing:

The AOCC software should meet the requirements of following modes of operation.

- Launcher phase
- South Sun Acquisition
- Transfer Orbit Earth Acquisition

- LAM Mode
- East/West Earth Acquisition
- Synchronous Orbit Earth Acquisition
- Station Keeping Mode
- On-Orbit Mode
- Safe Mode

Mode Post processing:

In this module, routines that are required to be executed after the completion of mode processing are done. Following modules are called in this portion.

- Quaternion Computation
- PWPFM Computation
- MW/RW Controller Computation
- MTC Observer Computation
- Actuator O/P Processing

Time Synchronization:

All the modules are to be executed once in a cycle. The cycle time is 32.768 msec. Once all the activities are completed in a cycle, this routine will be called to synchronize the time so that all the activities are synchronized with repetitive period of cycle time. This routine waits for a hardware timer output to become active and restarts the cycle and also reset the hardware timer.

All the new requirements of INSAT-2E were gathered and a draft copy of requirement document was prepared. A special expert committee reviewed the requirements for its completeness and requirements were validated. A software requirement document was prepared incorporating all the comments provided by the committee.

4.2 Design Phase

The software design carried out in earlier AOCCs was taken as a base line and the requirements are represented by flow diagrams. New requirements are incorporated in the flow diagram. Design analysis was carried out to optimize the flow diagrams. Here top down approach for software development was used. All the requirements are

decomposed into functional blocks and made as modules. Modules, which constitute one function, are combined under a package. Modular approach was followed and modules are segregated like computational bound module, data acquisition modules, actuator related processing modules, IO related module, etc. This type of decomposition gives reusability feature of software reuse for different types of spacecraft. Most of the computations are carried out in 32 bit floating point arithmetic. Whenever 32-bit precision was not sufficient, extended precision of 48 bit long float computations were used. Truncation errors in value accumulation were closely checked and proper care has been taken to adopt the single precision/extended precision computation, so that effect of truncation error is minimized. A design document was made and design review was carried out. Review committee recommendations were incorporated in the final software design document. Strict software configuration control was maintained by having reviews for any change incorporation. Any change used to be carried after proper reviews and modification in the document after getting approval from the configuration control committee.

4.3 Coding:

Before starting the AOCC software coding, a detailed coding guideline document was prepared, which was outcome of expert committee recommendations. This document was circulated to all the software code developers. Detailed guidelines are given in ref[9].

For AOCC software coding, bottom up approach was adopted. Here coding started with the bottommost unit. After coding, code review for the unit was carried out and review recommendations were incorporated. Then the unit was compiled using US-DOD approved TLD Ada compiler and unit was linked using TLD linker. ^{Testers} Stubs replaced all other modules and unit testing was carried out using TLD simulator with the help of TLD debugger. Unit testing consists of logic checks, flow checks, computational accuracy check for computation bound units. Same philosophy was used to code procedures, modules and packages. Proper documents were made by giving full details of test vectors, errors uncovered, types of error, corrective action taken etc. Suitable test scripts were written to provide test conditions, input values, since the

simulator does not support input/output operations. In this manner complete AOCC software was integrated & tested using TLD simulator with the help of TLD debugger except for I/O operations.

A team of experts carried out code walk-through using the Design document and Code document before final implementation. The comments/recommendation given by the code walk through committee were incorporated and the software review committee reviewed the full AOCC software. Software review committee recommendation was also incorporated in the AOCC S/W and Final version of the AOCC S/W was released for flight use.

4.4 Testing:

Testing starts with initial unit model development itself. Test plans were worked out from the S/W design document by giving full details of each and every test that are to be carried out to verify and validate the AOCC S/W. A test document was prepared by the Quality Assurance Team. After the final code development, the loaded module generated by the TLD linker are down loaded into the in-house development tool (DT) and the DT is interfaced with actual AOCC H/W. Here each module timing measurement was carried out. After getting the each module timing, modules are placed to meet the timing requirement. This completed the final software coding operation. After this the load module generated by TLD linker is converted into binary file for fusing into PROMs. Unisite Data IO programmer was used to fuse the S/W code into PROMS. Initially one set of EPROMS was fused and will be mounted onto the AOCC processor card. One test was conducted just to check the functionality. Once the AOCC is working in this configuration, the same S/W is fused into PROMS and PROMS are embedded inside AOCC processor card.

The final testing of AOCC is carried out using a Test System, which simulates all the interfaces the AOCC is having on the spacecraft. Tests are conducted as per test plan document. The tests conducted are Initial Bench Test - Functional verification, HILS test, Environmental Test and Integrated spacecraft Test.

5. Software Performance

The software had been extensively tested at ground before launch during various phases of testing. After launch, the S/C had been operated in almost all

modes like launcher phase, South Sun acquisition, Earth acquisition, Station keeping and normal On-orbit modes. No major problems have been observed. The time estimation was found satisfactory. No cases of computation overflows/underflows have been observed. However, there have been some feed backs on better monitoring parameters which can be upgraded in the future mission for better evaluation of the on orbit performance. There were certain observations during the S/W evaluation, which are needed to be considered in future. They are briefly discussed here.

- The timing overhead in calling a routine was very high. This has to be studied further.
- The I/O instruction's execution time of MIL-STD-1750 processor (MAR 281) was found to be very high. In future, design memory mapped I/Os can be thought off for better time performance.
- The scaling requirements on parameters are avoided by use of floating point variables. In continuous long run, single precision (32 bit) as well as extended precision (48 bit) (with truncation errors) may introduce inaccuracy. Therefore upper bound on each of the number need to be defined suitably.

6. Conclusion

The Attitude and Orbit Control Computer S/W of INSAT-2E is a mission critical software. This S/W is successfully designed, developed and used using MIL-STD-1750A processor with Ada language. A systematic, but simple development methodology has been used for S/W development. Though there were no problems found either on ground or on-board, it requires further improvement in reducing timing overhead and improving truncational inaccuracies. The AOCC is working continuously in INSAT-2E S/C since the day of launch i.e., 3rd April 1999.

References

1. "INSAT-2E AOCE Software Design Document", Internal report, Control Systems Group, ISRO Satellite Centre (ISAC), Bangalore.
2. "Hand book of AOCE H/W for INSAT-2E", Internal report, Control Systems Group, ISRO Satellite Centre (ISAC), Bangalore.

3. S. Satheesh, S.Manjunatha Prasanna, V.K. Agrawal, "Software Structure for IRS-1C Attitude and Orbit Control System", National Workshop on flight Software, Trivandrum, June 1996, pp 1.6 – 1.16.
4. V.K.Agrawal, M.M. Nayak, K.M. Bharadwaj, "Software Development for Attitude and Orbit Control Computer – A Methodology", National Workshop on flight Software, Trivandrum, June 1996, pp 2.28 – 2.43.
5. M. M. Nayak, R.S. Manjunath, V. K. Agrawal, N. K. Malik, " Software Development for Spacecraft Attitude and Orbit Control System (AOCS)", National Workshop on Software Engineering & its Applications, Institute of Public Enterprises, Hyderabad, Aug 1996.
6. S. Balaji, S.Manjunatha Prasanna, C. Rathna, V.K. Agrawal, Control Electronics Division, ISRO Satellite Centre, Bangalore-560 017 "Software Development for Spacecraft Onboard System Using a High level Language" - Paper published in the Journal of Spacecraft Technology, Vol. 6, No. 1, January 1996, pp 17 – 24.
7. Roger S. Pressman, "Software Engineering – A Practitioner's Approach, Fourth Edition, McGraw Hill International Edition, 1997.
8. "INSAT – 2E configuration document", Internal report, ISRO Satellite Centre (ISAC), Bangalore.
9. "Guidelines for On-board s/w development", Internal report, ISRO Satellite Centre (ISAC), Bangalore.
10. "INSAT – 2E AOCE onboard software source code", Internal report, Control Systems Group, ISRO Satellite Centre (ISAC), Bangalore.

Acknowledgement

The authors gratefully acknowledge the guidance provided by Dr. P.S Goel, Director, ISRO Satellite Centre, Bangalore during the course of this work.

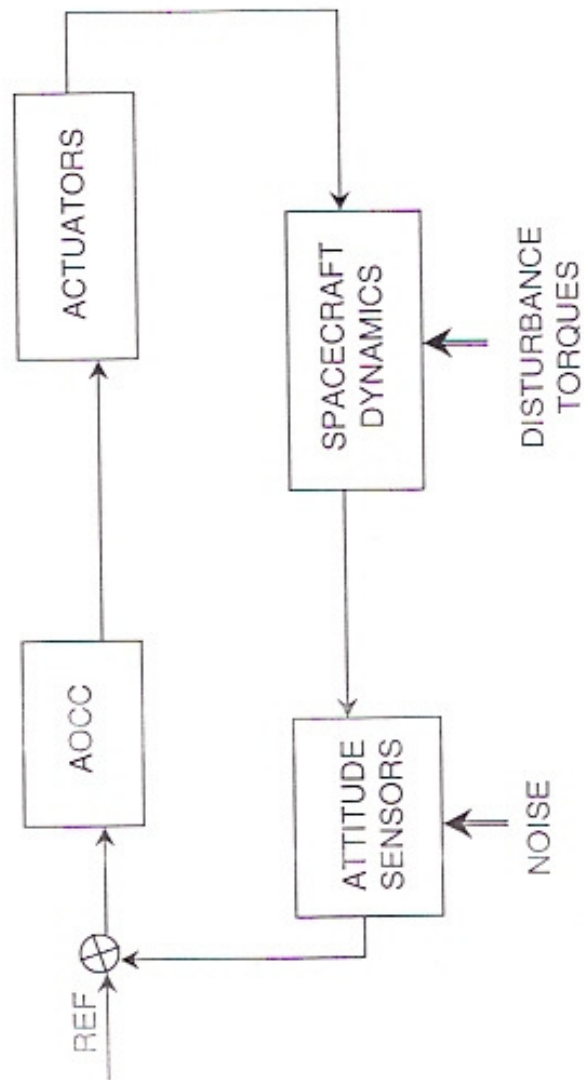


Fig. 1 : Block Diagram of Typical AOCS

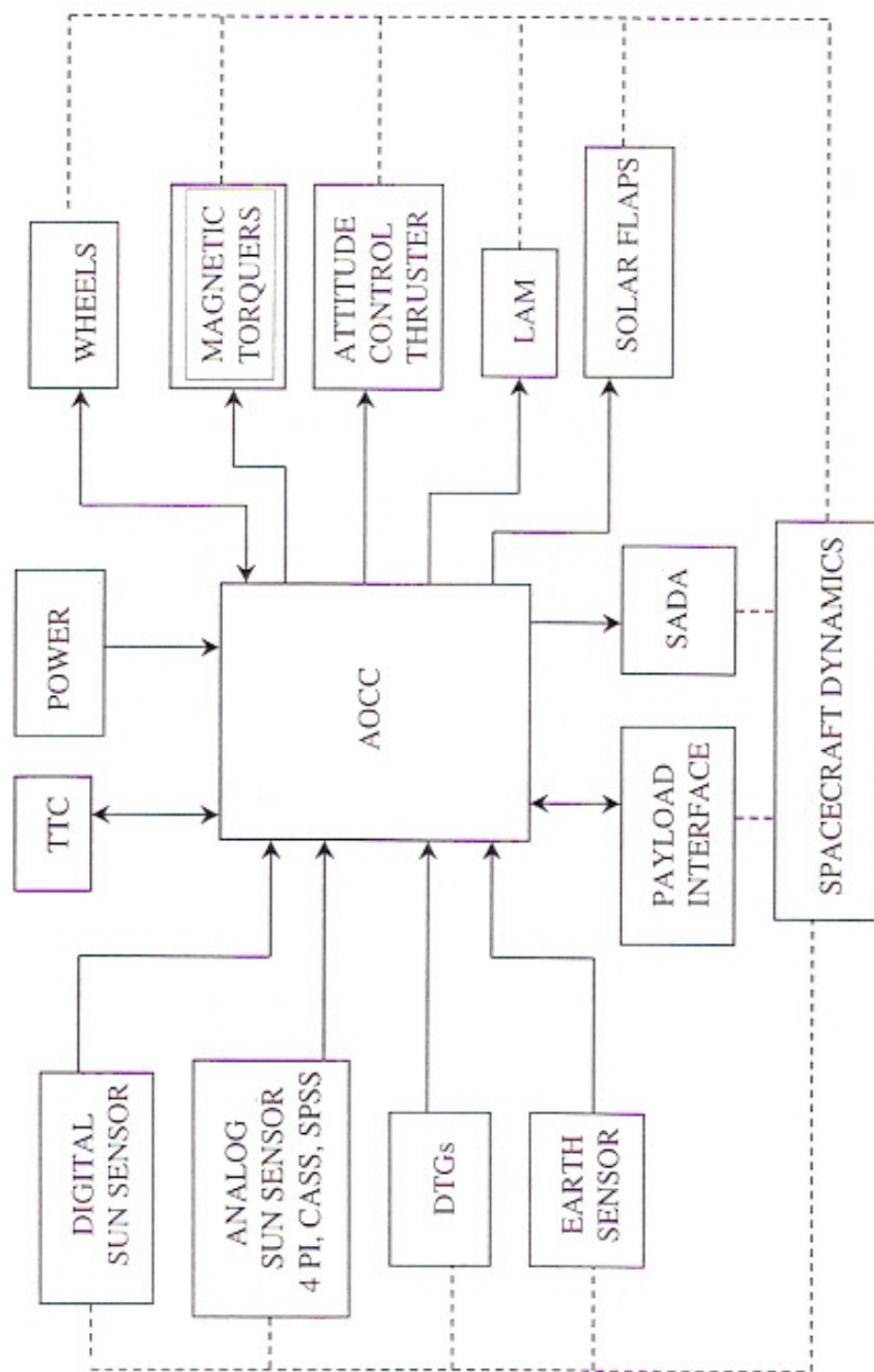


FIG. 2 : Block Schematic Giving AOCS Interfaces

MISSION PROFILE

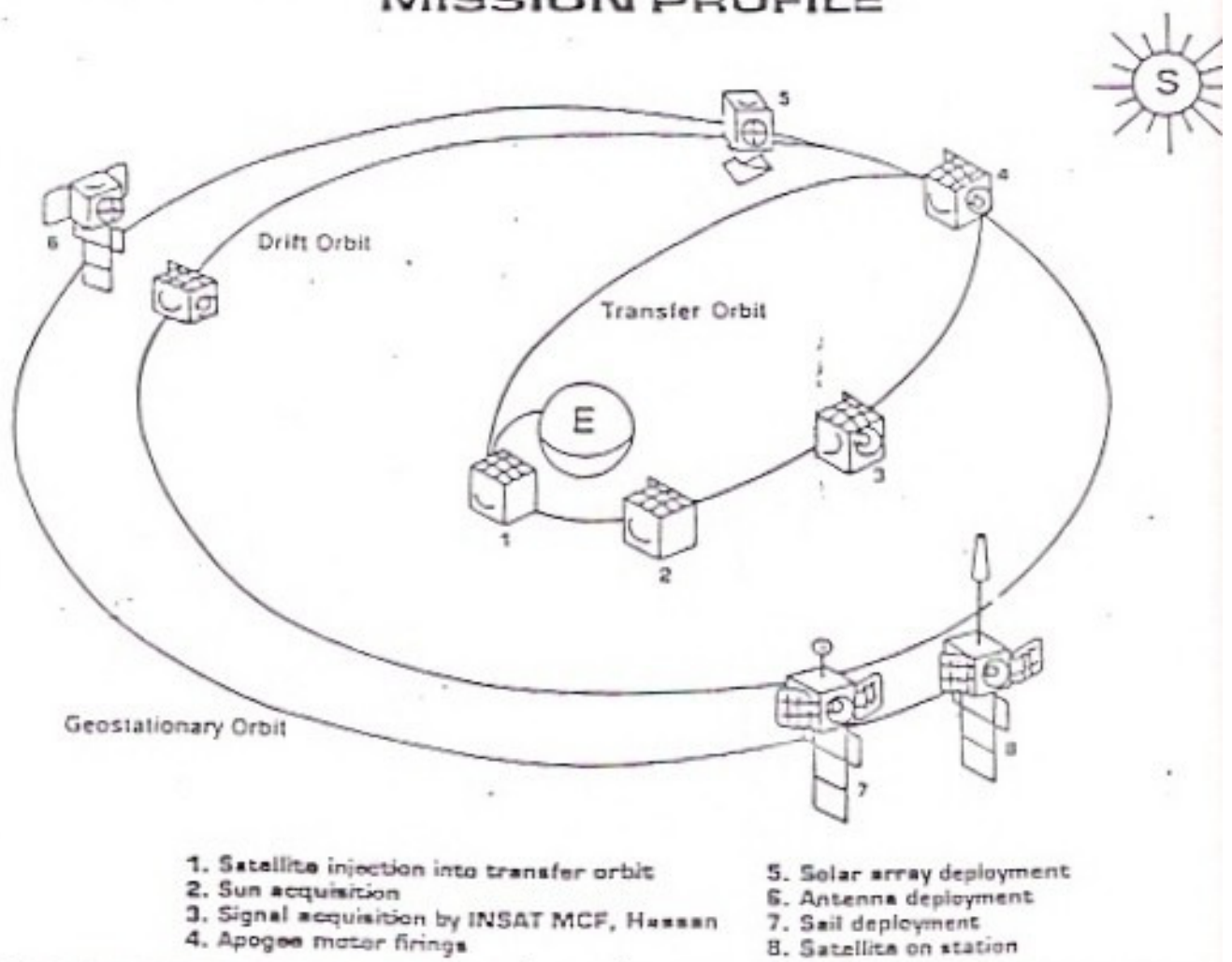


FIG. 3

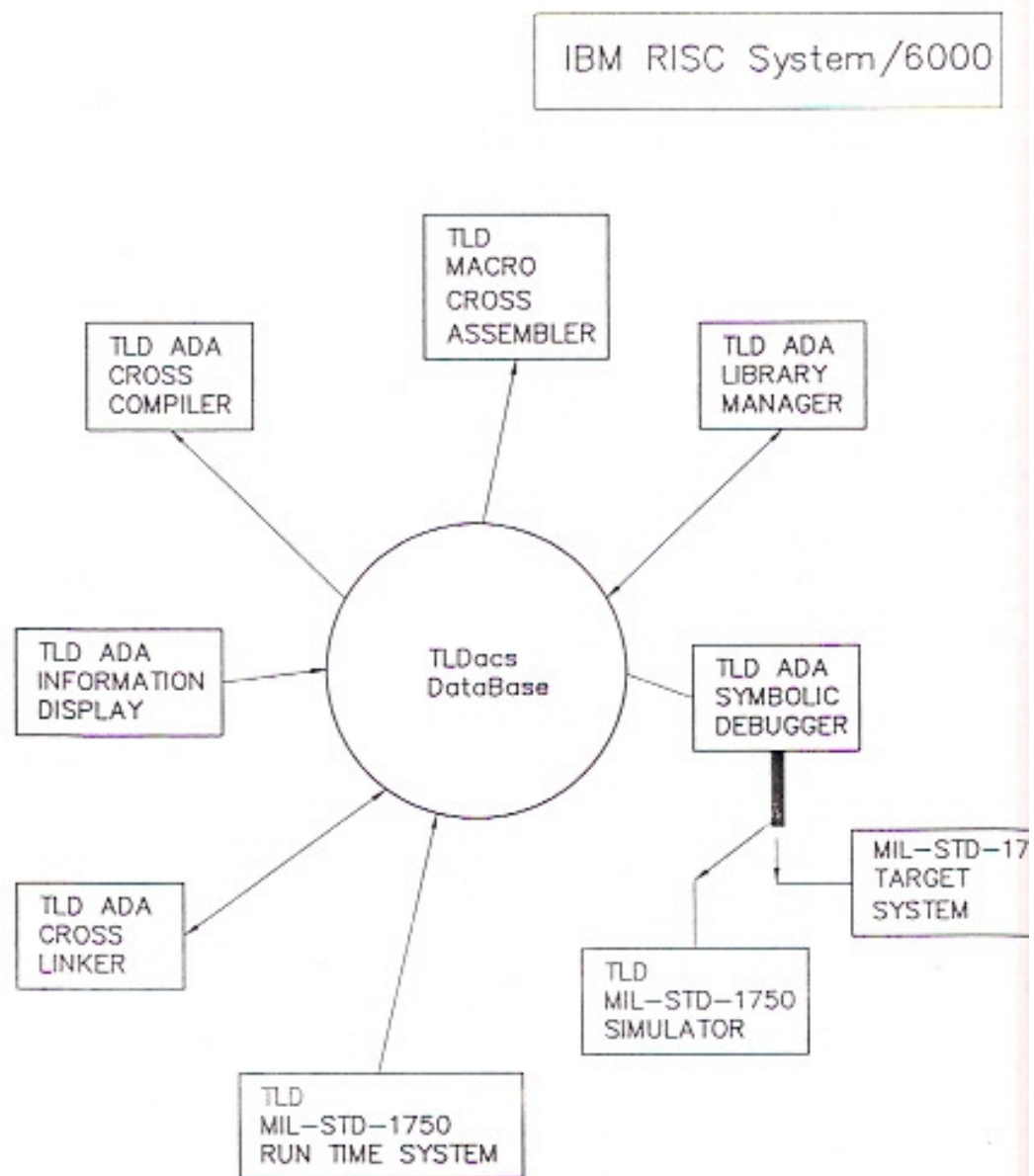


Fig.5 PROCESSOR DEVELOPMENT ENVIRONMENT

TABLE - 1

Sl.No.	S/W Routines	Execution Repetitive	Execution Ctrl Options	Options available	Remarks
1.0	PRE PROCESSING				
1.1	Launcher Phase	Once in Cycle	Enable/Disable	-	
1.2	TC Processing	- do -	-	-	
1.3	MTC Observer	- do -	Enable/Disable	Only in On-orbit mode	
1.4	Flap control	Twice in cycle	Enable/Disable	On-orbit, earth Acq. mode	
1.5	Solar array slewing	Once in cycle	-	On-orbit, earth Acq. mode	
1.6	TM Updating	- do -	-		
1.7	Orbit timer	- do -	-		
1.8	I/P port reading	- do -	-		
1.9	Sensor Data Processing	-	-		
1.10	Q propagation	Once in cycle	Enable/disable	All modes	
1.11	Tacho data processing	- do -	-		
2.0	Mode Specific Processing				
2.1	Rate damping Mode	Once in cycle	Mode command	Rate bias in all axis	

2.2	South sun acq. Mode	- do -	Mode command	4 Pi & DTG, CASS/DSS & DTG
2.3	T.O. Earth Acq. Mode	- do -	Mode command	4 Pi, ES, DTG; DSS, ES & DTG; IAC
2.4	LAM Mode	- do -	Mode command	ES3 & DTG; IAC
2.5	East-West Sun Acq. Mode	- do -	Mode command	4 Pi, CASS; IAC
2.6	S.O. Earth Acq. Mode	- do -	Mode command	4 Pi CASS; IAC
2.7	Station keeping Mode	- do -	Mode command	All DTG, ES & CASS, ES& DTG
2.8	On-orbit Mode	- do -	Mode command	ES, DTG acc. angle, Q's
2.9	Safe Mode	- do -	Mode command	
2.10	Suspended Mode	- do -	Mode command	
3.0	Post processing			
3.1	PWPFM computation	4 times in cycle	Enable/disable	All Acq. Modes
3.2	Thruster pattern processing	4 times in cycle	Enable/disable	All acq. modes
3.3	Momentum dumping	Once in a cycle	Enable/disable	Only in on-orbit mode
3.4	Auto transition	Once in a cycle	Enable/disable	Only in acq. mode
3.5	MW/RW controller	Once in a cycle	Enable/disable	Only in on-orbit mode
3.6	Wheel O/P processing	Once in a cycle	Enable/disable	
4.0	AOCE cycle completion check	Once in a cycle		